

Příznakové rozpoznávání

Feature-Based Recognition

Zadání bakalářské práce

Student: **Jakub Hartmann**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: **Příznakové rozpoznávání**
Feature-Based Recognition

Zásady pro vypracování:

Především v poslední době jsme svědky masivního nárůstu kamerových systémů. Tyto systémy mohou být primárně využívány pro zpracování a analýzu obrazu a to zejména pro klasifikaci objektů v obraze. Výsledná klasifikace objektů může být použita například pro bezpečnostní účely v dopravě, letectví, ale i v běžném životě každého z nás. V oblasti klasifikace objektů existuje skupina algoritmů, které jsou založeny na měření vlastností jednotlivých objektů. Tyto metody se nazývají příznakovými.

1. Seznamte se s pojmy souvisejícími s touto problematikou.
2. Následně se seznamte s existujícími přístupy pro rozpoznávání objektů v obrazech a jejich popis pomocí příznaků (příznakové rozpoznávání) tyto přístupy v práci popište.
3. Jednotlivé příznaky implementujte a proveďte testování.
4. V experimentální části uveďte výhody i nevýhody vybraných přístupů.

Seznam doporučené odborné literatury:

Podle pokynů vedoucího bakalářské práce.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Radovan Fusek**

Datum zadání: 18.11.2011

Datum odevzdání: 04.05.2012



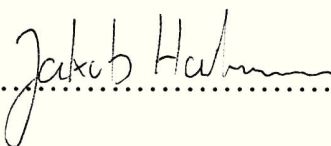
doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 23. dubna 2012

.....


Rád bych na tomto místě poděkoval Ing. Radovanu Fuskovi za vedení mé bakalářské práce, cenné rady a velkou trpělivost.

Abstrakt

Bakalářská práce s názvem "Příznakové rozpoznávání" se zabývá problematikou autorizovaného rozpoznávání objektů, vyskytujících se na obraze, na základě příznaků. Práce spadá do oblasti počítačového vidění. V teoretické části práce je popsán řetězec zpracování obrazu od jeho předzpracování až po klasifikaci. Zvýšený důraz je přitom věnován příznakům, jimž je věnována kapitola č.5. Praktická část je věnována implementaci vybraných příznaků s použitím knihovny OpenCV pro C++. Pro klasifikaci je pak použit rozhodovací strom, jehož vstupními daty jsou zjištěné příznaky.

Klíčová slova: zpracování obrazu, příznak, digitální obraz, klasifikace

Abstract

The thesis called "Feature-based recognition" deals with the issue of authorized recognition of objects occurring in an image and which is based on features. The thesis appertains to the field of computer vision. The theoretical part of the thesis describes a chain of image processing from its preprocessing up to the classification. The emphasis is put on features, which are dealt with in chapter 5. The practical part is dedicated to the implementation of selected features with the help of OpenCV library for C++. The detected features serve as input data for the classification by means of a decision tree.

Keywords: image processing, feature, digital image, classification

Obsah

1	Úvod	4
2	Základní pojmy a získávání obrazu	6
2.1	Digitalizace	6
2.2	Získávání obrazu	8
3	Předzpracování obrazu	10
4	Segmentace a identifikace oblastí	14
4.1	Prahování	14
4.2	Metody založené na detekci hran	16
4.3	Metody orientované na regiony	16
4.4	Identifikace oblastí	17
5	Popis objektu / Příznaky	19
5.1	Radiometrické příznaky	20
5.2	Fotometrické příznaky	23
5.3	Lokální příznaky	26
6	Klasifikace	27
6.1	Rozhodovací stromy	27
7	Praktická část	29
7.1	Tréninková data	29
7.2	Předzpracování	31
7.3	Získání příznaků	31
7.4	Testování	31
7.5	Zhodnocení	33
8	Závěr	35
9	Literatura	36
10	Přílohy	38

Seznam tabulek

1	Tabulka příznaků pro různě pootočený obdélník	32
2	Tabulka příznaků pro kruhy různých velikostí	33

Seznam obrázků

1	Ukázka několika objektů, s kterými se můžeme běžně setkat [14]	4
2	Záběry z autonomního automobilu firmy Google [15]	5
3	Ukázka digitalizace ručně psaného textu [13]	5
4	Typy vzorkovacích mřížek. Čtvercová, hexagonální a trojúhelníková [11]	6
5	Aditivní míchání barev modelu RGB [5]	7
6	Subtraktivní míchání barev u modelu CMY [5]	8
7	Kužel reprezentující model HSV [5]	8
8	Radiální zkreslení. a) Rastr b) Poduškovité zkreslení c) Soudkovité zkreslení [5]	10
9	Transformace jasové stupnice. a) Inverze b) Zvýšení kontrastu c) Prahování [8]	11
10	Původní obraz a jeho histogram (nahore), Obraz a jeho histogram po ekvalizaci (dole) [5]	11
11	Ukázka rozostření pomocí Gaussova filtru	12
12	Obraz po použití Cannyho detektoru hran [5]	13
13	Morfologické operace, uzavření a otevření (nahore), eroze a dilatace (dole), původní obrazec je tmavý, nový obrazec je světlý [5]	13
14	Ukázka originálního obrazu a binárního obrazu získaného prahováním [5]	14
15	Volba prahu T z histogramu [20]	15
16	Originální obraz. Obraz po segmentaci štěpením. Obraz po segmentaci spojováním a štěpením. [8]	17
17	Ukázka možného rozlišení oblastí pomocí čísel	17
18	Původní a prahovaný obraz (vlevo), první a druhý průchod algoritmu (vpravo) [8]	18
19	Získání příznakového vektoru [9]	19
20	Obdélník s velikostí 8317pxl a obvodem 384pxl. Hvězda s velikostí 3099pxl a obvodem 343pxl	20
21	Kruh s kompaktností 12.61 a konvexností 0.996. Kříž s kompaktností 28.29 a konvexností 0.711	21
22	Objekt (vlevo) a jeho konvexní obal (vpravo)	21
23	Objekt s Eulerovým číslem 1 (vlevo) a objekt s Eulerovým číslem 8 (vpravo)	22
24	Tvorba Freemanova kódu pro čtyř-okolí, respektive osmi-okolí [19]	22
25	Ukázka histogramů jednotlivých složek RGB a složeného histogramu [16]	24
26	Lokální příznaky [17]	26
27	Řetězec rozpoznávání z hlediska klasifikace	27
28	Ukazka rozhodovacího stromu s hloubkou 2	27
29	Příklad úlohy řešené rozhodovacím stromem [18]	28
30	Ukázka několika tréninkových obrázků	29
31	Diagram aplikace	30
32	Testovací obrazce	31
33	Kruhy použité pro testování	32
34	Obraz poškozený různým množstvím náhodného šumu	33

1 Úvod

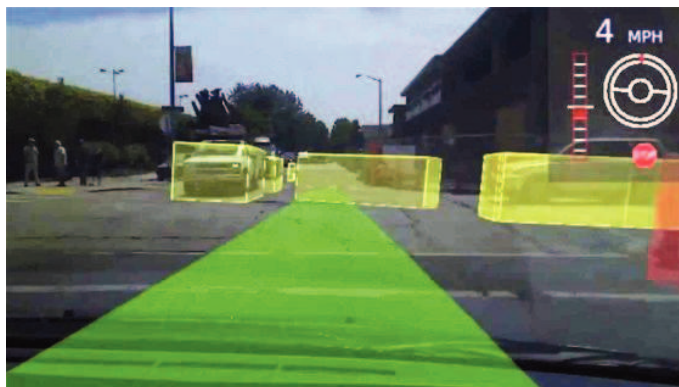
Počítačové vidění je obor, který se zabývá zpracováním a analýzou obrazu. Kloubí se v něm znalosti z různých oborů jako je například matematika, fyzika, informatika, umělá inteligence, neurobiologie apod. Specifickou částí počítačového vidění je rozpoznávání.



Obrázek 1: Ukázka několika objektů, s kterými se můžeme běžně setkat [14]

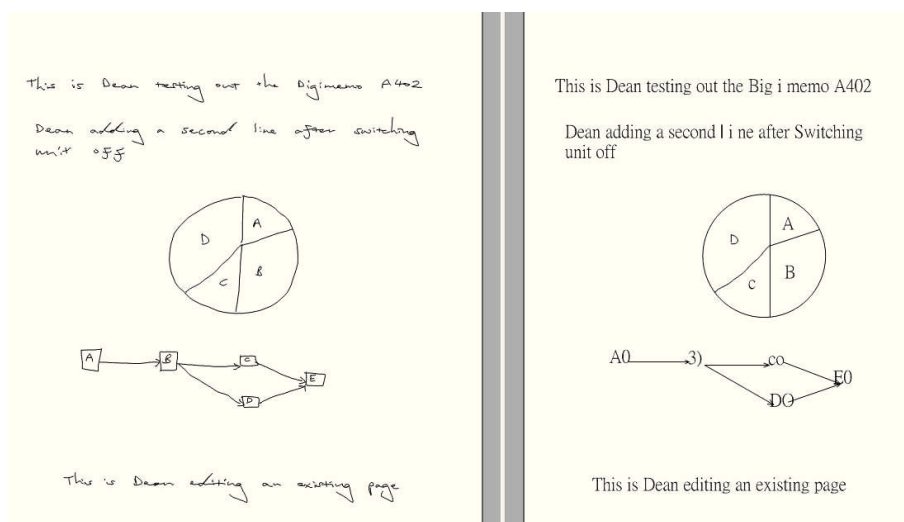
Pro člověka je běžné a snadné pomocí zraku rozpoznávat různé objekty, které se kolem něj nacházejí (viz Obr.1), ať již se jedná o lidi, zvířata či věci. Pokud ale chceme stejný problém řešit pomocí výpočetní techniky, stává se z něj poměrně složitý úkol. Aby byl počítač schopen různé předměty rozpoznat, musí být tyto objekty popsány pomocí pro ně specifických vlastností. V rámci počítačového vidění existuje celá skupina metod, které se zabývají popisem, rozpoznáváním a klasifikací právě na základě těchto vlastností tzv. příznaků. Tyto metody nazýváme příznakovými a můžeme je využít při řešení různých úloh v mnoha odvětvích.

Uvedme pro ilustraci možnost využití v automobilech. Může to být například systém pro automatické parkování vozidla, který je pomocí kamer schopen najít vhodné parkovací místo a automobil na něj navést bez asistence řidiče. Může se také jednat o systémy, které pouze pomáhají řidiči při jízdě a zvyšují bezpečnost. Zde můžeme uvést rozpoznávání dopravních značek nebo kontrolu dodržování jízdního pruhu. V neposlední řadě také existují systémy, které jsou schopné řídit auto v běžném provozu bez zásahu člověka (viz Obr.2).



Obrázek 2: Záběry z autonomního automobilu firmy Google [15]

Využití nalezneme také v různých OCR aplikacích, jako je například třídění poštovních zásilek na základě nasnímané adresy nebo digitalizace ručně psaných textů (viz Obr.3). A také v mnoha dalších oborech jako je medicína, vojenství, astronomie, geografie, defektoskopie apod.



Obrázek 3: Ukázka digitalizace ručně psaného textu [13]

V textu práce je rozebrána problematika získávání a zpracování obrazu a především je zde popsáno příznakové rozpoznávání. Práce je rozdělena do kapitol, z nichž první tři se zabývají pojmy, které souvisí s digitálním obrazem a jeho předzpracováním. Jsou zde popsány některé algoritmy a postupy, které mohou být při předzpracování využity. Kapitola 5 se pak zabývá popisem objektů a získáváním příznaků. Kapitola 6 je pak věnována klasifikaci. Poté následuje praktická část, v níž jsem implementoval některé příznaky, provedl jejich testování a zhodnotil výsledky.

2 Základní pojmy a získávání obrazu

2.1 Digitalizace

Obraz (obrazová data) lze reprezentovat jako spojitě rozložené jasů (intenzity, optické hustoty) v rovině. [6] Můžeme jej popsat pomocí reálné, spojitě obrazové funkce o dvou proměnných, pokud se jedná o statická data.

$$f(x, y)$$

V případě obrazů proměnných v čase má tato funkce 3 proměnné (x a y představují bod v rovině, t představuje hodnotu v čase)

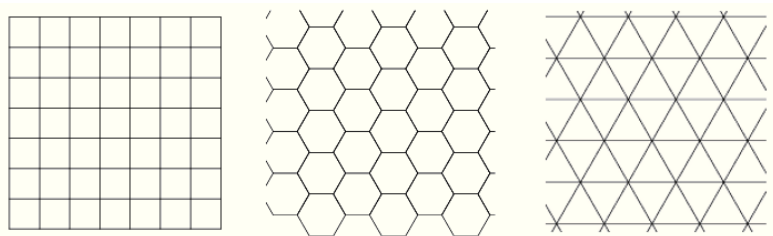
$$f(x, y, t)$$

Takto je reprezentován pouze šedotónový obraz. Pokud je navíc obor hodnot omezen pouze na dvě hodnoty, jedná se o binární obraz. Pro barevný obraz použijeme několik funkcí, pomocí kterých budou vyjádřeny jednotlivé složky barevného modelu např. RGB (HSV, YUV, CMY ...)

$$f_R(x, y); f_G(x, y); f_B(x, y)$$

Pro práci s obrazem v počítači použijeme digitalizovaný obraz, který je vyjádřen pomocí matice $M \times N$, jejíž jednotlivé prvky nazýváme pixely. Pixel je nejmenší a dále již nedělitelný element digitálního obrazu nesoucí informaci o odpovídajícím obrazovém bodě. Hodnoty jednotlivých pixelů jsou získány vzorkováním obrazové funkce. Pro každý vzorek je hodnota kvantována do jedné z K úrovní. Tím získáme konečný počet hodnot, které jsou obvykle vyjádřeny pomocí celých čísel.

Čím vyšší je rozlišení (počet pixelů) a počet kvantovacích úrovní, tím přesnější je reprezentace původního obrazu. Při vzorkování platí Shannonův teorém. „Přesná rekonstrukce spojitěho, frekvenčně omezeného signálu z jeho vzorků je možná tehdy, pokud byl vzorkován frekvencí alespoň dvakrát vyšší, než je maximální frekvence rekonstruovaného signálu.“ [5] Příliš nízké rozlišení vede ke ztrátě detailů obrazu, naopak příliš vysoké rozlišení zvyšuje výpočetní náročnost dalších operací prováděných s obrazem. Volba vhodného rozlišení je tedy velmi důležitá. Kromě rozlišení je také nutné zvolit vzorkovací mřížku (viz Obr.4) a také počet kvantovacích úrovní. Při nízkém počtu kvantovacích úrovní dochází ke vzniku falešných obrysů, které jsou pro lidské oko pozorovatelné, pokud je počet úrovní nižší než cca. 64 (u monochromatického obrazu).



Obrázek 4: Typy vzorkovacích mřížek. Čtvercová, hexagonální a trojúhelníková [11]

2.1.1 Reprezentace barev

Důležitým parametrem obrazu je jeho barva. Všechny pixely obrazu obsahují informaci o jednotlivých složkách použitého barevného modelu. Nejběžnější barevné modely jsou

- RGB
- CMY
- HSV

RGB je aditivní model, což znamená, že výsledná barva je dána součtem tří základních barev (R - červená, B - modrá, G - zelená). Pokud jsou všechny tři barvy zastoupeny maximální intenzitou, získáme bílou barvu, naopak pokud mají všechny nulovou intenzitu, získáme barvu černou. Tento model se používá u většiny zobrazovacích zařízení jako jsou televize, monitory, projektory apod. Někdy se také můžeme setkat s modelem RGBA, který přidává navíc informaci o průhlednosti pixelu tzv. alfa kanál (A).



Obrázek 5: Aditivní míchání barev modelu RGB [5]

CMY je model subtraktivní. V tomto případě je výsledná barva získána tak, že jednotlivé barvy azurová (C), purpurová (M) a žlutá (Y) jsou od sebe odečítány. Modely RGB a CMY lze mezi sebou velmi snadno převádět podle následujícího vzorce.

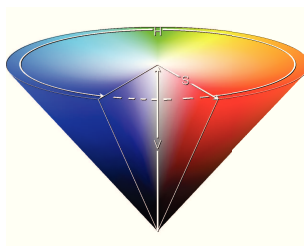
$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Tento model se používá obvykle v tiskárnách, kde obvykle přidáváme navíc černou barvu (K) a dostáváme tak model CMYK.



Obrázek 6: Subtraktivní míchání barev u modelu CMY [5]

HSV (někdy také HSB) je model, který nejlépe odpovídá lidskému vnímání barev. Jeho složky jsou barevný tón (H), sytost barvy (S) a hodnota jasu (V). Tento model je využíván některými grafickými aplikacemi. Jeho použití může být v některých případech výhodné při segmentaci.



Obrázek 7: Kužel reprezentující model HSV [5]

2.2 Získávání obrazu

Z fyzikálního hlediska můžeme snímání obrazu chápat jako převod vstupní veličiny na elektrický signál. Vstupní veličinou může kromě jasu být například také intenzita rentgenového záření, tepelná intenzita, ultrazvuk atd. V následujícím textu ale budeme jako vstupní veličinu uvažovat vždy jas. Řetězec získávání a zpracování obrazu má několik kroků.

- Snímání
- Digitalizace
- Předzpracování
- Segmentace
- Popis objektů
- Klasifikace

Pro získání obrazu určité scény, je nutné, aby byla daná scéna osvětlena. To může být zajištěno buď denním světlem, nebo přídavnými zdroji světla. Ty můžeme rozdělit například podle vlnové délky vyzařovaného světla na ultrafialové, viditelné a infračervené. Podle způsobu osvětlení dané scény pak na bodové a plošné. Vhodným typem a intenzitou osvětlení ovlivníme především viditelnost snímaného objektu a také můžeme zvýraznit některé jeho požadované vlastnosti.

Snímací zařízení, v našem případě fotoaparát, má dvě hlavní části. První částí je objektiv. Jedná se o soustavu optických čoček, která má za úkol soustředit světelné paprsky na senzor fotoaparátu. Základními vlastnostmi objektivu jsou ohnisková vzdálenost a světelnost. U některých objektivů je možné ohniskovou vzdálenost v určitém rozsahu měnit (tzv. zoom). Podle ohniskové vzdálenosti můžeme rozdělit na širokoúhlé (úhel záběru větší než 75°), normální (úhel záběru kolem 50°) a teleobjektivy (úhel záběru menší než 30°). U objektivů se vyskytují vady, které mají vliv na výsledný obraz. Tyto vady můžeme rozdělit do následujících skupin

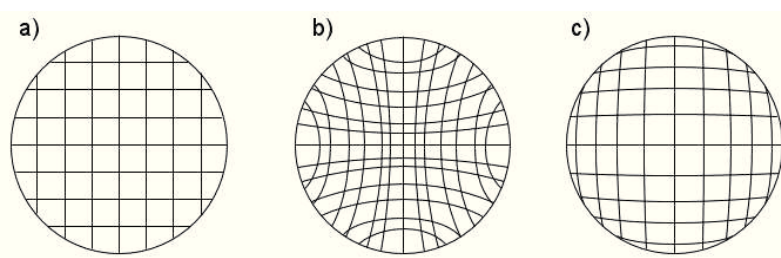
- sférické aberace
- chromatické aberace
- astigmatismy
- asymetrické aberace
- zkreslení obrazu (soudkovité, poduškovité)
- sklenutí zorného pole
- difrakční vady
- vinětace atd.

Druhou částí je senzor, který převádí dopadající světlo na elektrický signál. Podle použité technologie dělíme senzory na CCD (Charge-coupled device) a CMOS (complementary metal oxide semiconductor). V dnešní době se v běžných fotoaparátech využívá většinou technologie CMOS, která poskytuje dostatečné rozlišení, odolnost proti šumu a především nízkou cenu.

3 Předzpracování obrazu

Při předzpracování nám jde především o zlepšení nebo potlačení určitých vlastností obrazu tak, aby ho poté bylo možné lépe využít v dalších fázích zpracování. Pokud víme, jaké operace chceme dále s obrazem provádět, popřípadě pokud víme, jaká z něj chceme získat data, můžeme při předzpracování použít takové operace, abychom dosáhli nejlepších výsledků. Mezi možné vylepšení patří odstranění šumu a zkreslení, nalezení hran, zvýšení kontrastu a další.

Šum obvykle vzniká digitalizací nebo přenosem obrazu. Můžeme jej rozdělit na několik typů např. bílý, Gaussův, "pepř a sůl", aditivní. Také zkreslení je možné rozdělit do několika skupin - radiální (poduška, soudek) (viz Obr.8), tangenciální, změna měřítka, atd. Při odstraňování těchto chyb využíváme toho, že jednotlivé pixely jsou obvykle velmi podobné svému blízkému okolí a pro výpočet hodnoty určitého pixelu můžeme použít hodnoty pixelů v jeho blízkém okolí.



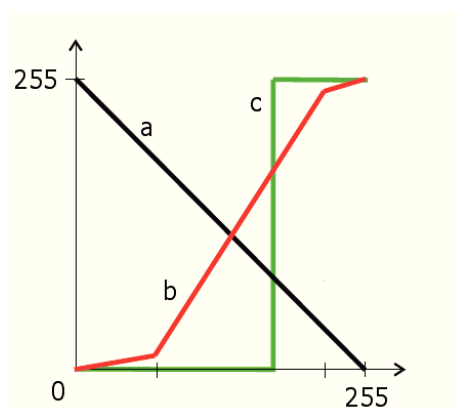
Obrázek 8: Radiální zkreslení. a) Rastr b) Poduškovité zkreslení c) Soudkovité zkreslení [5]

Jednou z možných úprav při předzpracování obrazu jsou bodové jasové transformace. Jas v bodě výstupního obrazu závisí pouze na jasu bodu ve vstupním obrazu = pro úpravu jednoho konkrétního pixelu použijeme jen tento pixel vstupního obrazu. [12] Příkladem takovéto transformace mohou být například jasová korekce

$$I_C(i, j) = M \frac{I_0(i, j) - I_b(i, j)}{I_f(i, j) - I_b(i, j)}$$

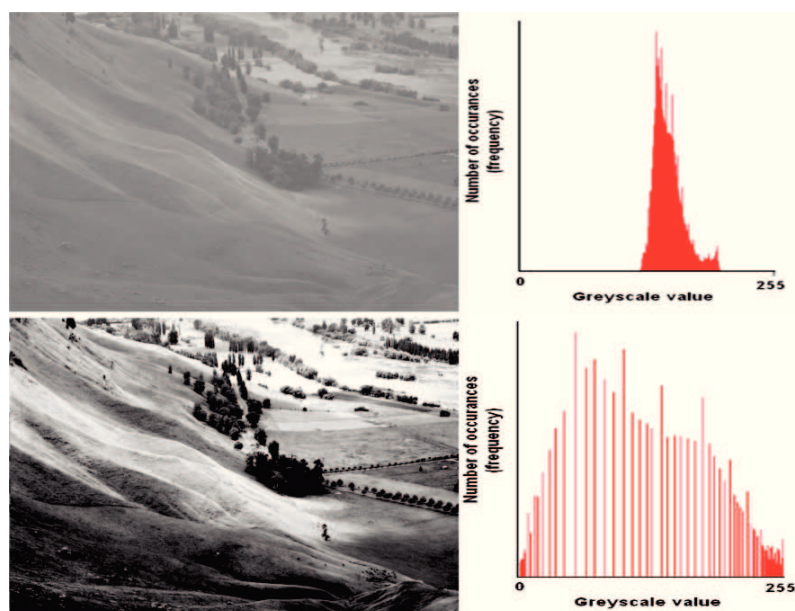
kde I_0 je obraz bez objektu, I_b je obraz pořízený bez osvětlení, I_f slouží pro korekci nelinearity snímače a pomocí konstanty M měníme kontrast obrazu.

Další úpravou jsou například transformace jasové stupnice, kam může zařadit například inverzi, ekvalizaci histogramu, úpravu kontrastu nebo redukci barev.



Obrázek 9: Transformace jasové stupnice. a) Inverze b) Zvýšení kontrastu c) Prahování [8]

Pomocí geometrických transformací odstraňujeme vady obrazu popsané výše. Tyto úpravy můžeme rozdělit do dvou fází. První z nich je plošná transformace, po níž následuje jasová transformace. Pomocí plošné transformace nalezneme polohu bodů z původního obrazu v obrazu upraveném. Abychom byli schopni nalézt polohu nových bodů, je nutné znát transformační vztahy, podle kterých bude výpočet proveden. Ty jsou buď předem známy, nebo jsou určeny z informací o původním a upraveném obraze. Pomocí jasové transformace pak nalezneme pro body v transformovaném obraze hodnoty jejich jasu.



Obrázek 10: Původní obraz a jeho histogram (nahore), Obraz a jeho histogram po ekvalizaci (dole) [5]

Jak je patrné již z názvu, u lokálních metod předzpracování se pro výpočet hodnot pixelů v upraveném obraze využívá okolí pixelu v obraze původním. Tyto metody jsou obvykle využity pro vyhlazení obrazu nebo pro nalezení hran. Pro vyhlazení můžeme použít například průměrování, rotující masku nebo Gaussův filtr.

$$\frac{1}{256} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

Při vyhlazení obrazu průměrováním obvykle dochází k rozmazání hran, tomu se můžeme vyhnout pomocí průměrování přes několik obrazů.



Obrázek 11: Ukázka rozostření pomocí Gaussova filtru

$$h_1 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}, h_2 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{pmatrix}, h_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, h_4 = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$

$$h_5 = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}, h_6 = \begin{pmatrix} 3 & 3 & 3 \\ 3 & 0 & 3 \\ -5 & -5 & -5 \end{pmatrix}$$

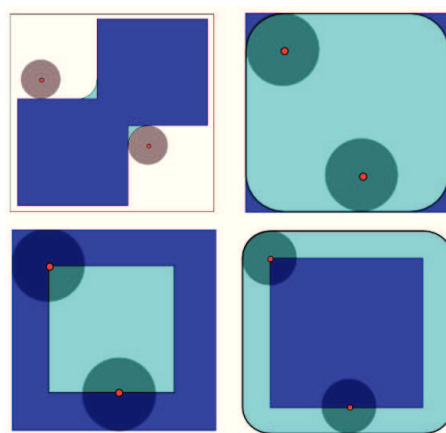
Zde jsou ukázky několika konvolučních masek, které se používají pro detekci hran. h_1 a h_2 jsou Laplaceovy masky, dále pak následují masky Robertsova, Sobelova, Prewitova a Kirsch. Při zvýraznění hran se potýkáme s problémem, že hrany mají stejně jako šum vysoký gradient, a tudíž při zvýraznění hran dochází také ke zhoršení šumu.



Obrázek 12: Obráz po použití Cannyho detektoru hran [5]

Při restauraci obrazu se snažíme získat zpět obrazová data, která byla porušena. Čím větší je naše znalost o poruše, která ztrátu dat způsobila, tím lepších výsledků můžeme při restauraci dosáhnout. Informace o poruše jsou nám buďto předem známy nebo je možné je získat analýzou poškozeného obrazu. Příkladem restaurace obrazu může být např. Wienerův nebo Kalmanův filtr.

Matematická morfologie se používá k odstranění šumu, zjednodušení tvaru objektů nebo popisu objektů. Původně byla vyvinuta pro binární obrazy, následně ale byla rozšířena tak, aby ji bylo možné použít i pro šedotónové obrazy. Patří sem například operace jako dilatace, eroze, otevíření, uzavření, ztenčování, zesilování apod.



Obrázek 13: Morfologické operace, uzavření a otevíření (nahore), eroze a dilatace (dole), původní obrazec je tmavý, nový obrazec je světlý [5]

4 Segmentace a identifikace oblastí

Cílem segmentace je rozdělení obrazu do částí, které korespondují s objekty z reálného světa. Po provedení segmentace by měl být obraz připraven pro další fázi zpracování a tou je popis. V nejlepším případě dosáhneme úplné segmentace, což znamená, že vysegmentované oblasti korespondují s jednotlivými objekty v obraze. Bohužel není vždy možné rozlišit jednotlivé objekty od pozadí. Oblasti vzniklé segmentací potom přesně neodpovídají jednotlivým objektům (tzv. částečná segmentace). Pro segmentaci je možné využít například globální znalosti jako je barva, tvar, jas, poloha atd.

4.1 Prahování

Prahování (angl. Thresholding) je zřejmě nejjednodušší metodou pro segmentaci obrazu. V určitých případech se však jedná o velmi rychlou a spolehlivou metodu. Výstupem prahování je binární obraz. Prahování je založeno na myšlence, že hledané objekty mají stejnou, nebo podobnou odrazivost a tudíž jednotlivé pixely, náležící danému objektu, mají velmi podobné hodnoty jasu nebo barvy. Pro správnou segmentaci je nutné, aby hledané objekty měly vlastnosti rozdílné od pozadí. Pro rozhodnutí, zda daný pixel je či není objektem, slouží hodnota zvaná práh. Uvažujeme-li prahování na základě jasu, pak v nejjednodušším případě, kdy máme pouze jeden práh T , jsou pixely, jejichž jas je vyšší než hodnota prahu T , označeny jako objekt a pixely, které mají jas nižší než daný práh T jsou označeny jako pozadí, případně naopak. Tato transformace se dá zapsat následujícím vztahem.

$$G(i, j) = \begin{cases} 1 & \text{pro } I(i, j) \geq T \\ 0 & \text{pro } I(i, j) < T \end{cases}$$



Obrázek 14: Ukázka originálního obrazu a binárního obrazu získaného prahováním [5]

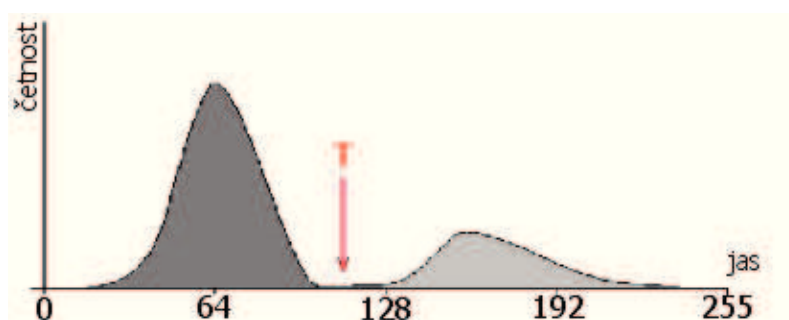
Rozlišujeme několik druhů prahování.

- **Prosté** - Máme pouze jeden práh T , podle nějž je prahován celý obraz. Způsobů, jak práh určit, je hned několik. Některé z nich jsou uvedeny níže.

- **S více prahy** - Můžeme mít větší počet prahů. Výsledný obraz pak již není binární, ale dostáváme obraz s přesně definovaným počtem jasových úrovní.
- **Částečné** - Pixelům, jejichž jas je vyšší než práh T , je přiřazena bílá barva. Pixely, jejichž jas je nižší než práh T , jsou na rozdíl od běžného prahování ponechány beze změny. Tento druh prahování může být použit například u obrazů, které pozoruje a hodnotí člověk.
- **Adaptivní** - Hodnota prahu je určena jednotlivě pro menší části obrazu. Tímto způsobem můžeme dosáhnout lepší segmentace například u obrazů, které jsou nerovnoměrně osvětleny

Dále pak existuje několik přístupů k určení prahu.

- **Experimentální**
- **Z histogramu** - Tato metoda funguje velice dobře, pokud je histogram obrazu bimodální, tedy pokud má dvě od sebe dobře odlišitelná maxima. V ostatních případech nemusí tato metoda poskytovat uspokojivé výsledky. Pokud máme vhodný histogram, můžeme snadno určit práh například jako poloviční vzdálenost mezi maximy nebo jako lokální minimum mezi dvěma maximy apod.
- **Procentní** - Vycházíme z předpokladu, že hledané objekty nebo oblasti pokrývají určité procento plochy snímku. Pokud předpokládáme, že hledaný objekt zabírá 30% snímku, určíme práh T tak, aby právě 30% plochy histogramu mělo jasovou úroveň vyšší, než určený práh T .
- **Ze statistik** - Práh T je určen na základě určité statistiky vypočtené z dané oblasti. Například to může být medián, střední hodnota, $(\max + \min)/2$ apod.
- **Z globálních znalostí** - Pokud máme určitou znalost hledaného objektu, můžeme vycházet z této vlastnosti. Příkladem takové vlastnosti může být barva kůže.



Obrázek 15: Volba prahu T z histogramu [20]

4.2 Metody založené na detekci hran

Existuje několik způsobů, jak nalézt hranu. Prvním z nich je použití některého z detektorů hran při předzpracování obrazu (Sobel, Prewit, Kirch, Roberts atd.). Dalším způsobem je určení hrany jako krajních pixelů určité oblasti (například na základě jasu). Také můžeme využít určité informace, které o objektu máme, jako je například jeho tvar nebo barva, a využít těchto znalostí k nalezení hrany, případně k ověření správnosti nalezené hrany. Při hledání hran máme několik požadavků, které by měly být splněny. Za prvé by neměla být vynechána žádná významná hrana nebo naopak by neměla být za hranu označena žádná oblast, která hranou není. Dalším požadavkem je požadavek na jednoznačnost, což znamená, že každé skutečné hraně odpovídá pouze jedna nalezená hrana. Posledním požadavkem je požadavek na přesnost. Rozdíl mezi skutečnou a nalezenou hranou by měl být co nejmenší, ideálně nulový.

Metody založené na detekci hran jsou následující:

- prahování obrazu hran
- sledování hranice
- heuristické sledování hranice
- aktivní kontury
- atd.

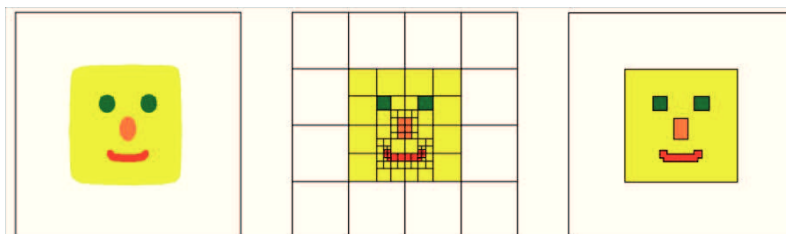
4.3 Metody orientované na regiony

Segmentace narůstáním oblastí Tato metoda je založena na rozdělení vstupního obrazu do menších částí a posléze na jejich postupném spojování do větších homogenních oblastí. Při rozhodování, zda mají být dvě dané oblasti spojeny, se můžeme řídit například pomocí jasu, barvy, textury apod. Tato metoda může poskytovat různé výsledky a to na základě různých počátečních oblastí, pořadí oblastí, spojovací podmínky atd. Její algoritmus je následující.

1. Rozděl vstupní obraz na velký počet podoblastí (čím jsou jednotlivé oblasti menší, tím je výsledek lepší, ideální je rozdělení na jednotlivé pixely).
2. Vytvoř podmínku pro spojení dvou sousedících podoblastí.
3. Pokud dvě sousedící podoblasti splňují podmínku pro spojení, pak je spoj. Pokud již neexistují žádné dvě oblasti, které by bylo možné spojit bez toho, aby byla porušena podmínka, skonči.

Segmentace štěpením oblastí Postup u této metody je opačný než u metody předchozí. Začínáme s celým vstupním obrazem a ten v jednotlivých krocích dělíme na menší podoblasti do té doby, dokud jednotlivé podoblasti nesplňují zadanou podmínku. Při použití metody štěpení oblastí nemusíme dojít ke stejným výsledkům jako u metody spojování.

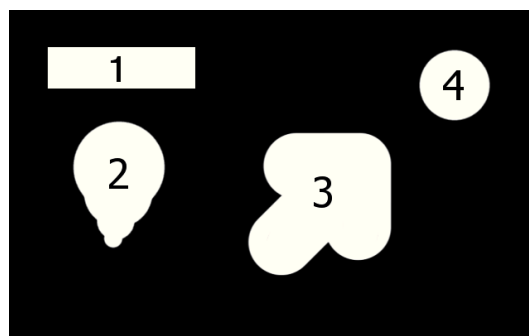
Segmentace štěpením a spojováním Jak je patrné již z názvu, jedná se o kombinaci postupů použitých u předchozích dvou metod. Pokud je daná oblast nehomogenní, dojde k jejímu rozdělení do čtyř podoblastí. Naopak, jsou-li oblasti navzájem homogenní, dojde k jejich spojení do jedné.



Obrázek 16: Originální obraz. Obraz po segmentaci štěpením. Obraz po segmentaci spojováním a štěpením. [8]

4.4 Identifikace oblastí

Po segmentaci bychom měli v nejjednodušším případě dostat binární obraz, který obsahuje neidentifikované oblasti. Účelem tohoto kroku je označení, a tedy rozlišení jednotlivých spojitých oblastí (rozpoznávaných objektů) tak, abychom s nimi byli schopni dále pracovat. Pro označení oblastí můžeme použít například celá čísla. Při zjišťování, zda spolu pixely sousedí a tvoří tedy spojitou oblast, rozlišujeme čtyř- a osmi-sousednost. Při čtyř-sousednosti jsou dva pixely sousedící, pokud mají společnou stranu. Při osmi-sousednosti spolu dva pixely sousedí nejen v případě, že mají společnou stranu, ale i v případě, kdy mají společný pouze roh.



Obrázek 17: Ukázka možného rozlišení oblastí pomocí čísel

Jednoduché semínkové vyplňování - Algoritmus semínkového vyplňování je následující

1. Otestuj, zda bod již nebyl vyplněn nebo neleží na hranici.
2. Pokud ano, skonči, pokud ne, vyplň tento bod a urči všechny sousední body.

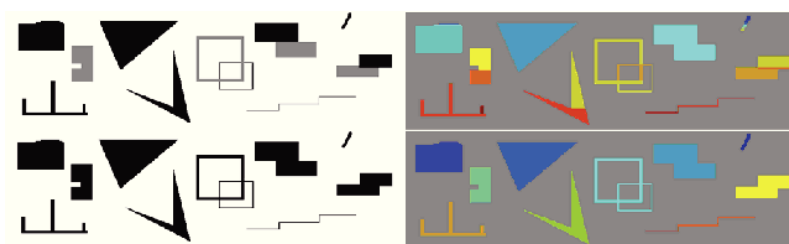
3. Pro všechny sousední body znovu použij tento algoritmus.

Jelikož je tento algoritmus paměťově náročný (každý pixel se testuje několikrát, mnohonásobně zanořená rekurze), používá se jeho upravená verze, kde se rekurze používá pouze ve směrech nahoru a dolů a na řádcích se využívá sekvenční vyplnění - Řádkové semínkové vyplňování.

Dvou-průchodový algoritmus Je další možností, kterou lze využít pro rozpoznání jednotlivých souvislých ploch v segmentovaném obraze. Pro označení jednotlivých oblastí můžeme použít 1 až N přirozených čísel, pro pozadí je obvykle použita číslice 0 a ostatním oblastem jsou poté postupně přiřazována ostatní čísla. Každá oblast je označena svým unikátním číslem.

Jiný způsob, který je možné použít, využívá pouze omezený počet číslic (teoreticky stačí použít čtyři) tak, aby žádné dvě sousední oblasti neměly stejné číslo. Aby bylo poté možné jednotlivé oblasti od sebe rozpoznat, je nutné nějakým způsobem zaznamenat jejich polohu, což můžeme provést například zaznamenáním polohy jednoho bodu, který náleží dané oblasti. Pro jedna až N číslic je algoritmus následující.

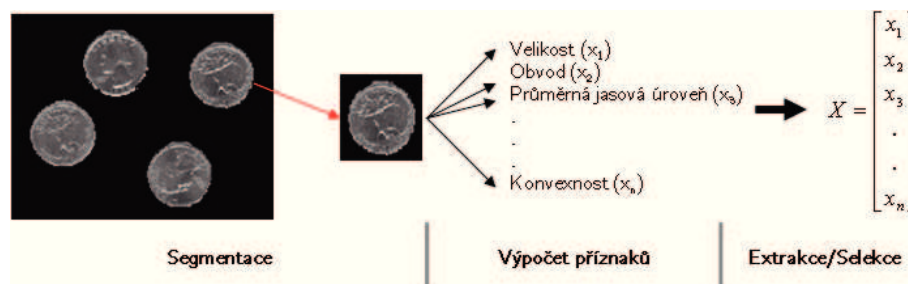
1. Procházíme postupně celý obraz a každý dosud neoznačený nenulový pixel označíme číslem, které mají pixely v jeho okolí.
 - (a) Pokud jsou všechny okolní pixely označeny číslem nula, označíme bod dosud nepřiděleným číslem.
 - (b) Pokud je jeden nebo více okolních pixelů označeno jedním číslem, označíme bod právě tímto číslem.
 - (c) Pokud je více okolních pixelů označeno různými čísly, pak bod označíme jedním z nich a uděláme záznam o kolizi indexů do tabulky ekvivalence.
2. Znovu projdeme celý obraz a podle záznamů o kolizích indexů opravíme značení oblastí.



Obrázek 18: Původní a prahovaný obraz (vlevo), první a druhý průchod algoritmu (vpravo) [8]

5 Popis objektu / Příznaky

Základem popisu objektu je získání takových vlastností (příznaků) daného objektu, které jsou pro daný objekt specifické a umožňují tedy jeho dobrou klasifikaci. Získané příznaky tvoří tzv. příznakový vektor, který může obsahovat libovolný počet příznaků. Naší snahou však je, aby počet příznaků byl co nejmenší, neboť s každým dalším příznakem se zvyšuje výpočetní náročnost. Ke zmenšení vektoru můžeme použít následující postupy.



Obrázek 19: Získání příznakového vektoru [9]

Extrakce příznaků - redukuje dimenzi příznakového prostoru tak, že transformuje původně získané příznaky na menší počet nových příznaků. Nově vytvořené příznaky mají odlišný význam od původních příznaků. Možnou metodou použitelnou k extrakci je například Karhunen-Loevova transformace (KLT).

Selekce příznaků - z původní množiny příznaků se vybírá podmnožina příznaků, které vykazují nejlepší schopnost rozlišení mezi jednotlivými třídami objektů (tzn. mají nejvyšší diskriminabilitu). Na rozdíl od extrakce se u selekce význam jednotlivých příznaků nemění, ale přicházíme o část informací o objektu v důsledku nepoužití některých příznaků.

Příznaky by měly splňovat následující požadavky:

- Invariantnost - příznak není závislý na změně jasu, kontrastu, měřítka, rotaci, translaci
- Spolehlivost - příznaky objektů jedné třídy mají obdobné hodnoty
- Diskriminabilita - příznaky pro různé třídy se od sebe rozlišitelné
- Efektivita výpočtu - příznak je dobře a rychle detekovatelný
- Časová invariance - příznak má stabilní hodnotu při zpracování dynamických obrazů

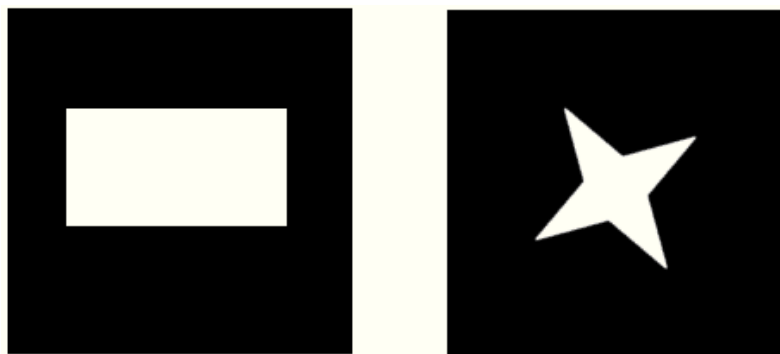
5.1 Radiometrické příznaky

Popisují objekt za pomoci jeho geometrických vlastností. Vycházejí z tvaru a velikosti daného objektu. Je tedy velice důležité, aby jednotlivé objekty byly co nejpřesněji segmentovány.

5.1.1 Příznaky založené na regionech

Velikost - Je dána celkovým počtem pixelů, které náleží danému objektu. Pokud použijeme normalizační konstantu, pak se při zvyšování rozlišení obrazu blíží příznak velikosti skutečné velikosti objektu.

Obvod - Je dán počtem pixelů, které tvoří hranici daného objektu. Obvod je možno určit pro čtyř nebo osmi okolí. Pokud pro osmi-okolí použijeme koeficient $\sqrt{2}$, pak je hodnota pro 4 i 8 okolí rovnocenná. Hodnota tohoto příznaku se s rostoucím rozlišením blíží nekonečnu.



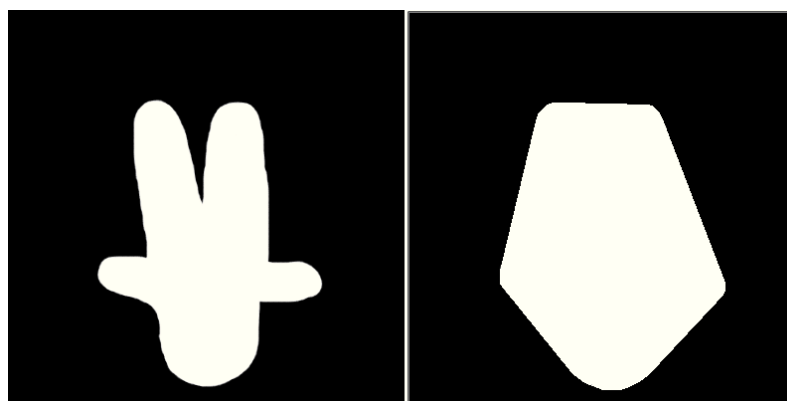
Obrázek 20: Obdélník s velikostí 8317pxl a obvodem 384pxl. Hvězda s velikostí 3099pxl a obvodem 343pxl

Kompaktnost - Je definována jako $Obvod^2 / Velikost$. Určuje podobnost s ideálním kruhem, jehož kompaktnost je rovna 4π . Čím je hodnota kompaktnosti nižší, tím je podobnost daného objektu s ideálním kruhem větší a naopak.



Obrázek 21: Kruh s kompaktností 12.61 a konvexností 0.996. Kříž s kompaktností 28.29 a konvexností 0.711

Konvexnost - Je definována jako *Velikost/Plocha konvexního obalu*. Konvexní obal objektu je takový, pro který platí, že úsečka spojující dva libovolné body objektu, leží uvnitř tohoto obalu (viz Obr.22). Možné hodnoty tohoto příznaku se pohybují mezi 0 a 1. Příznak nabývá hodnoty 1 pro konvexní (vypouklé) objekty."



Obrázek 22: Objekt (vlevo) a jeho konvexní obal (vpravo)

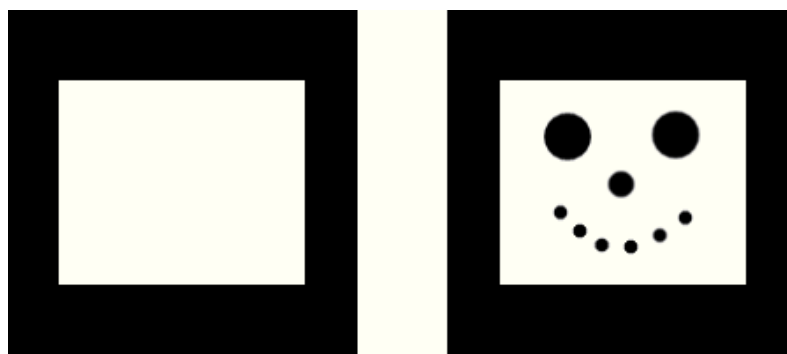
Hlavní osa - Délka hlavní osy elipsy, jejíž moment druhého řádu odpovídá centrálnímu momentu druhého řádu daného objektu.

Vedlejší osa - Délka vedlejší osy elipsy, jejíž moment druhého řádu odpovídá centrálnímu momentu druhého řádu daného objektu.

Podlouhlost - Je dána poměrem stran obdélníku opsaného danému objektu, přičemž musí být splněna podmínka, že obdélník musí mít minimální možný obsah.

Pravouhlost - Je dána maximálním poměrem velikosti daného objektu a plochy opsaného obdélníka. Stejně jako u předchozího příznaku, i zde musí mít obdélník minimální možnou plochu. Tento příznak může nabývat hodnot z intervalu $< 0, 1 >$.

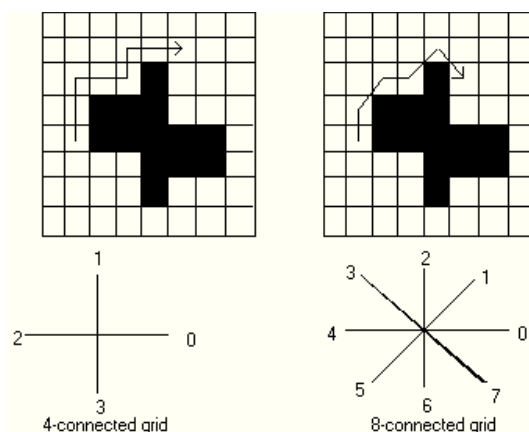
Eulerovo číslo (genus) - Je vyjádřeno rozdílem počtu souvislých ploch objektu a počtu děr v objektu. Tento příznak je invariantní vůči geometrickým transformacím obrazu.



Obrázek 23: Objekt s Eulerovým číslem 1 (vlevo) a objekt s Eulerovým číslem 8 (vpravo)

5.1.2 Příznaky založené na hranicích

Freemanův kód Pro popsání hranice objektu je u Freemanova neboli řetězového kódu použita posloupnost symbolů (nejčastěji čísel). Počet použitých symbolů je buď 4 pro čtyř-okolí, nebo 8 pro osmi-okolí. Obvykle se pro jednotlivé směry používají číslice nula až tři pro čtyř-okolí, respektive nula až sedm pro osmiokolí, není to však pravidlem. Je vhodné vždy popsat, podle jakých pravidel bude Freemanův kód vytvářen, aby se předešlo chybám.



Obrázek 24: Tvorba Freemanova kódu pro čtyř-okolí, respektive osmi-okolí [19]

5.2 Fotometrické příznaky

Fotometrické příznaky jsou, na rozdíl od příznaků radiometrických, založeny na jasových hodnotách segmentovaných dat. Pro výpočet tedy není možné použít binární obraz, ale je nutné použít segmentovaná data, obsahující jasové hodnoty jednotlivých pixelů. K fotometrickým příznakům patří například následující.

Průměrná jasová úroveň objektu (Ω)

$$B_{mean} = \frac{1}{N} \sum_{(x,y) \in \Omega} f(x,y)$$

Minimální a maximální jasová úroveň objektu

$$B_{min} = \min(f(x,y)), B_{max} = \max(f(x,y)), (x,y) \in \Omega$$

Diference minimální a maximální jasové úrovně

$$B_{range} = B_{max} - B_{min}$$

Diference jasové úrovně objektu Ω a okolí Φ

$$B_{diff} = B_{mean(\Omega)} - B_{mean(\Phi)}$$

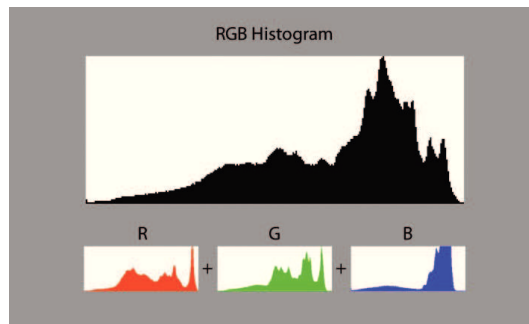
Parametry histogramu Histogram je z hlediska statistiky grafickým znázorněním distribuce dat pomocí sloupcového grafu. Sloupce představují jednotlivé intervaly a výška sloupců vyjadřuje četnost sledované veličiny v daném intervalu. V případě jasového histogramu máme na horizontální ose obvykle 256 jasových hodnot od 0 (černá) po 255 (bílá) a na svislé ose máme vyneseno, jakou plochu (počet pixelů) jednotlivé hodnoty v daném obraze zaujímají. Jasový histogram můžeme lehce sestavit jak ze šedotónového obrazu, tak z obrazu barevného. U barevného obrazu se pro výpočet jasu nabízí jednoduchý vzorec

$$\text{Absolutní jas} = 1/3R + 1/3G + 1/3B$$

Jelikož je však lidské oko jinak citlivé na každou složku RGB modelu, použijeme upravený vzorec

$$\text{Jas} = 0.3R + 0.59G + 0.11B$$

Samozřejmě je také možné sestavit samostatné histogramy pro červenou, zelenou i modrou složku modelu RGB.



Obrázek 25: Ukázka histogramů jednotlivých složek RGB a složeného histogramu [16]

Zde je několik ukázek parametrů, které můžeme z histogramu získat:

Průměrná hodnota

$$H_{mean} = \frac{1}{N} \sum_{q=1}^N q \cdot h(q)$$

Kontrast

$$H_{con} = \frac{1}{N} \sum_{q=1}^N (q \cdot h(q) - H_{mean})^2$$

Energie

$$H_{energy} = \frac{1}{N} \sum_{q=1}^N h(q)^2$$

Entropie

$$H_{entropy} = \frac{1}{N} \sum_{q=1}^N h(q) \cdot \log_2 h(q)$$

Geometrické momenty a invarianty Pro popis objektů lze také velmi dobře využít příznaky zvané geometrické momenty. Jejich výpočet je poměrně nenáročný, ale zároveň je jejich diskriminabilita dosti vysoká. Geometrické momenty jsou fotometrickými příznaky, které jsou založeny na regionech, což znamená, že jejich hodnota je závislá jak na tvaru daného objektu, tak na jasových hodnotách jeho pixelů. Pro jakýkoliv objekt je možné sestavit neomezený počet momentů, které jej budou popisovat. S rostoucím počtem momentů je možné daný objekt přesněji popsat a rozlišit jej od jiných objektů. Ovšem u momentů vyšších řádů musíme počítat s tím, že se u nich projeví šum.

Základní geometrický moment řádu $p+q$ vzhledem k souřadným osám obrazu je dán následujícím vztahem, kde Ω je definičním oborem obrazové funkce $f(x, y)$

$$m_{pq} = \int \int_{\Omega} x^p y^q f(x, y) dx dy$$

Tento vztah je platný pouze pro spojitou obrazovou funkci. Pokud chceme provést výpočet momentů pro funkci diskrétní, nahradíme integraci sumací a dostaneme následující vztah, podle nějž momenty počítáme.

$$m_{pq} = \sum_Y \sum_X x^p y^q f(x, y)$$

Jelikož momenty, s výjimkou momentů nultého a prvního řádu, vypočtené vzhledem k osám obrazu, jsou ovlivněny polohou daného objektu v obraze, je vhodnější provést výpočet momentů vzhledem k osám procházejícím těžištěm objektu. Ovšem pokud je poloha objektu pro jeho klasifikaci důležitá, můžeme použít jako příznak i základní momenty. Polohu těžiště je možno vypočítat za použití následujících vzorců. Pro souřadnici těžiště x_t a y_t

$$x_t = \frac{m_{10}}{m_{00}}, \quad y_t = \frac{m_{01}}{m_{00}}$$

Polohu těžiště je posléze možné využít pro výpočet centrálních momentů a to podle vzorce.

$$\mu_{pq} = \sum_Y \sum_X (x - x_t)^p \cdot (y - y_t)^q \cdot f(x, y)$$

Centrální moment μ_{pq} je nezávislý na poloze objektu, stále je však závislý na velikosti. Abychom se této závislosti zbavili, je nutné použít normalizované geometrické momenty, které je možné vypočítat za pomoci mocniny nultého momentu dle následujícího vzorce.

$$v_{pq} = \frac{m_{pq}}{m_{00}^{\frac{p+q}{2}+1}}$$

Pokud bychom požadovali také nezávislost tohoto deskriptoru na obecné afinní transformaci, musíme použít tzv. momentové invarianty, které jsou vypočteny z normalizovaných momentů. Mezi nejběžnější patří následující sada sedmi invariantních momentů [9].

$$\Phi_1 = \mu_{20} + \mu_{02}$$

$$\Phi_2 = (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2$$

$$\Phi_3 = (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2$$

$$\Phi_4 = (\mu_{30} + \mu_{21})^2 + (\mu_{21} + \mu_{03})^2$$

$$\Phi_5 = (\mu_{30} - 3\mu_{12}) \cdot (\mu_{30} + \mu_{12}) \cdot [(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] + \\ (3\mu_{21} - \mu_{03}) \cdot (\mu_{21} + \mu_{03}) \cdot [3(\mu_{30} + \mu_{21})^2 - (\mu_{21} + \mu_{03})^2]$$

$$\Phi_6 = (\mu_{20} - \mu_{02}) \cdot [(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] + 4\mu_{11} \cdot (\mu_{30} + \mu_{21}) \cdot (\mu_{21} + \mu_{03})$$

$$\Phi_7 = (3\mu_{21} - \mu_{03}) \cdot (\mu_{30} + \mu_{12}) \cdot [(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] - \\ (\mu_{03} - 3\mu_{12}) \cdot (\mu_{21} + \mu_{03}) \cdot [3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2]$$

5.3 Lokální příznaky

Jako významný bod můžeme označit ty části obrazu, které jsou velmi odlišné od svého okolí. Je to místo, ve kterém se mění několik vlastností obrazu současně. Je možné takové místa opakovaně nalézt v transformovaném obraze. Jedná se tedy zejména o vrcholy, rohy a hranice. K detekci významných bodů můžeme použít například Moravcův nebo Harrisův operátor, laplacián gausiánů, diferenciál gausiánů. Při hledání významných bodů není nutná sémantická segmentace.



Obrázek 26: Lokální příznaky [17]

6 Klasifikace

Účelem klasifikace je zařazení zkoumaných objektů do příslušných tříd na základě získaných příznaků. Při klasifikaci je důležité vybrat vhodnou množinu příznaků, podle kterých budeme objekty rozdělovat. Samotné rozdělování je provedeno klasifikačním algoritmem, tzv. klasifikátorem. Klasifikátory můžeme rozdělit například podle použitých metod na symbolické, subsymbolické, statistické a paměťové. Dále je můžeme rozdělit na klasifikátory používající učení s učitelem nebo bez učitele.

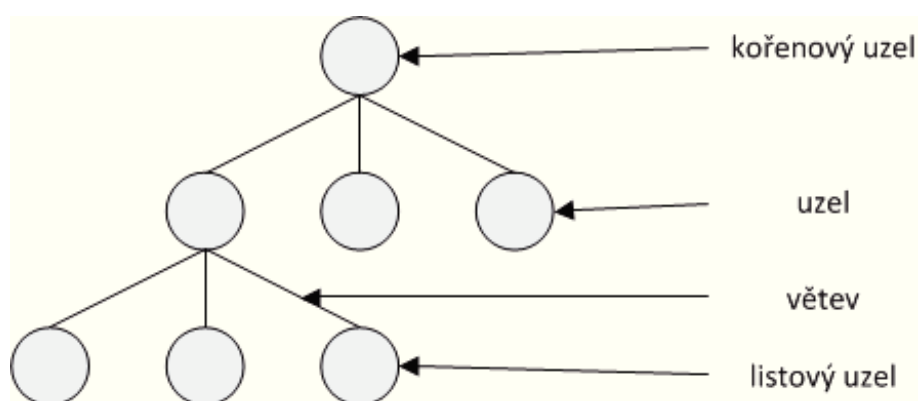


Obrázek 27: Řetězec rozpoznávání z hlediska klasifikace

6.1 Rozhodovací stromy

Rozhodovací stromy jsou jedny z velmi používaných klasifikátorů a to především pro jejich jednoduchou interpretovatelnost pro člověka, možnost použití pro různé typy dat, schopnost zpracovat i neúplná data. Rozhodovací stromy můžeme rozdělit podle topologie (binární, vícerozměrné), nebo podle vstupní a výstupní proměnné (kvalitativní, kvantitativní).

Tento klasifikátor je možné reprezentovat, jak už je patrné z názvu, pomocí stromové struktury. Jednotlivé nelistové uzly představují podmínky (v případě obrazových dat se může jednat např. o hodnotu jasu, barvu), podle nichž se strom dále větví až k listovým uzlům, které představují jednotlivé třídy objektů. Pro kořenový uzel je vhodné vybrat vlastnost objektů, která má velkou separabilitu. A v dalších uzlech postupovat k méně důležitým vlastnostem. Separabilitu dané vlastnosti určuje entropie. Dále je důležité omezit větvení stromu, aby nedošlo k příliš podrobnému rozčlenění.

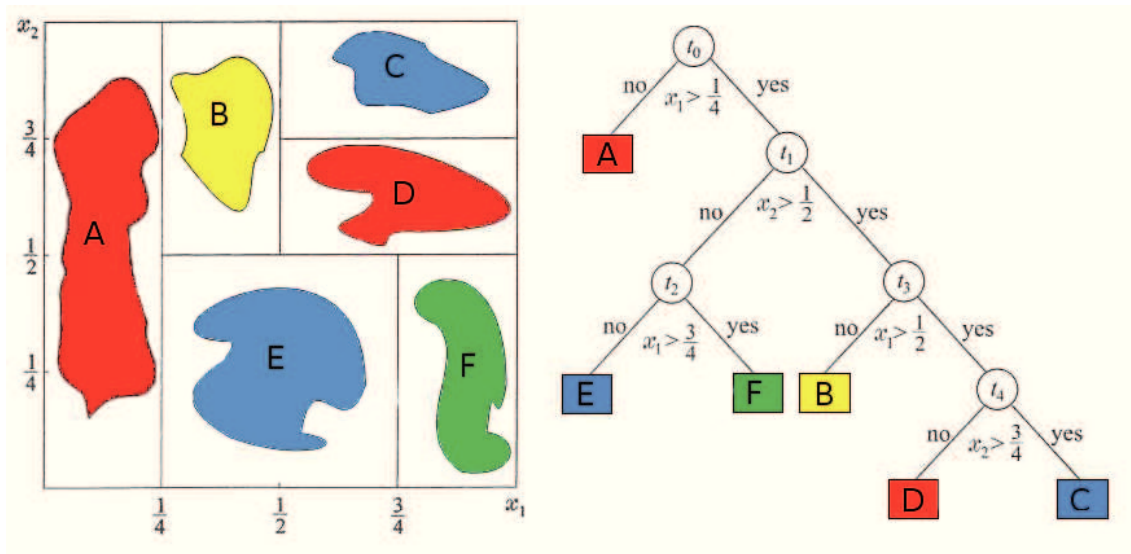


Obrázek 28: Ukazka rozhodovacího stromu s hloubkou 2

CART (Classification And Regression Tree) Algoritmus je založen na počítání míry diverzity ("nečistoty") v každém uzlu stromu. Pro určení míry diverzity se používá Giniho index [1].

$$div_{gini} = 1 - \sum p_i^2$$

Kde p_i jsou relativní četnosti v uzlech. Tento algoritmus tvoří pouze binární stromy. Většinou necháme vyrůst strom do maximální šíře, což však vede k přeučení, a poté strom prořežeme, čili odstraníme listy a větve, které nelze považovat za významné (rozhodujeme podle statistického kritéria, obvykle cross-validation).



Obrázek 29: Příklad úlohy řešené rozhodovacím stromem [18]

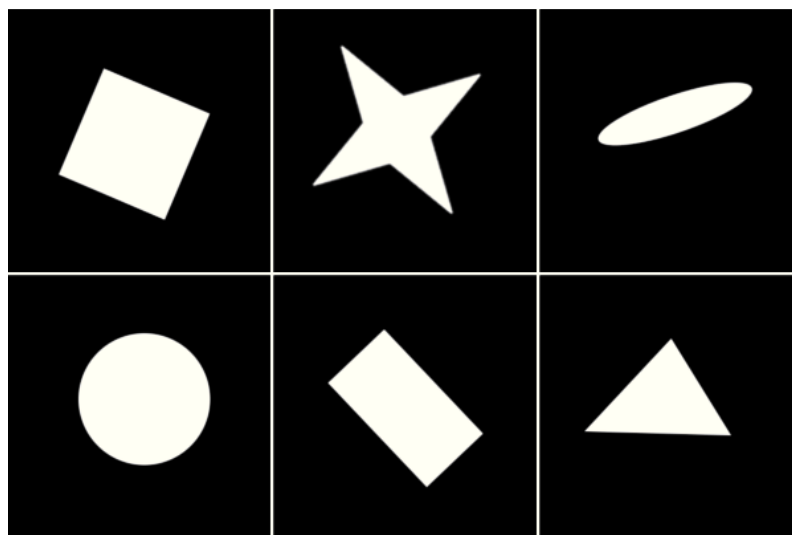
7 Praktická část

Před započítím realizace praktické části bakalářské práce bylo nutné zvolit vhodné programovací prostředí. Zvolil jsem programovací jazyk C++ společně s open source knihovnou pro zpracování obrazu, OpenCV a to v nejvyšší dostupné verzi 2.3.1 (duben 2012). Knihovna OpenCV obsahuje velké množství optimalizovaných funkcí, jak pro práci se samotným obrazem, tak také funkcí z oblasti strojového učení určené pro klasifikaci. Od verze 2.0 podporuje OpenCV také rozhraní C++. Její použití tedy velmi usnadňuje práci s obrazem.

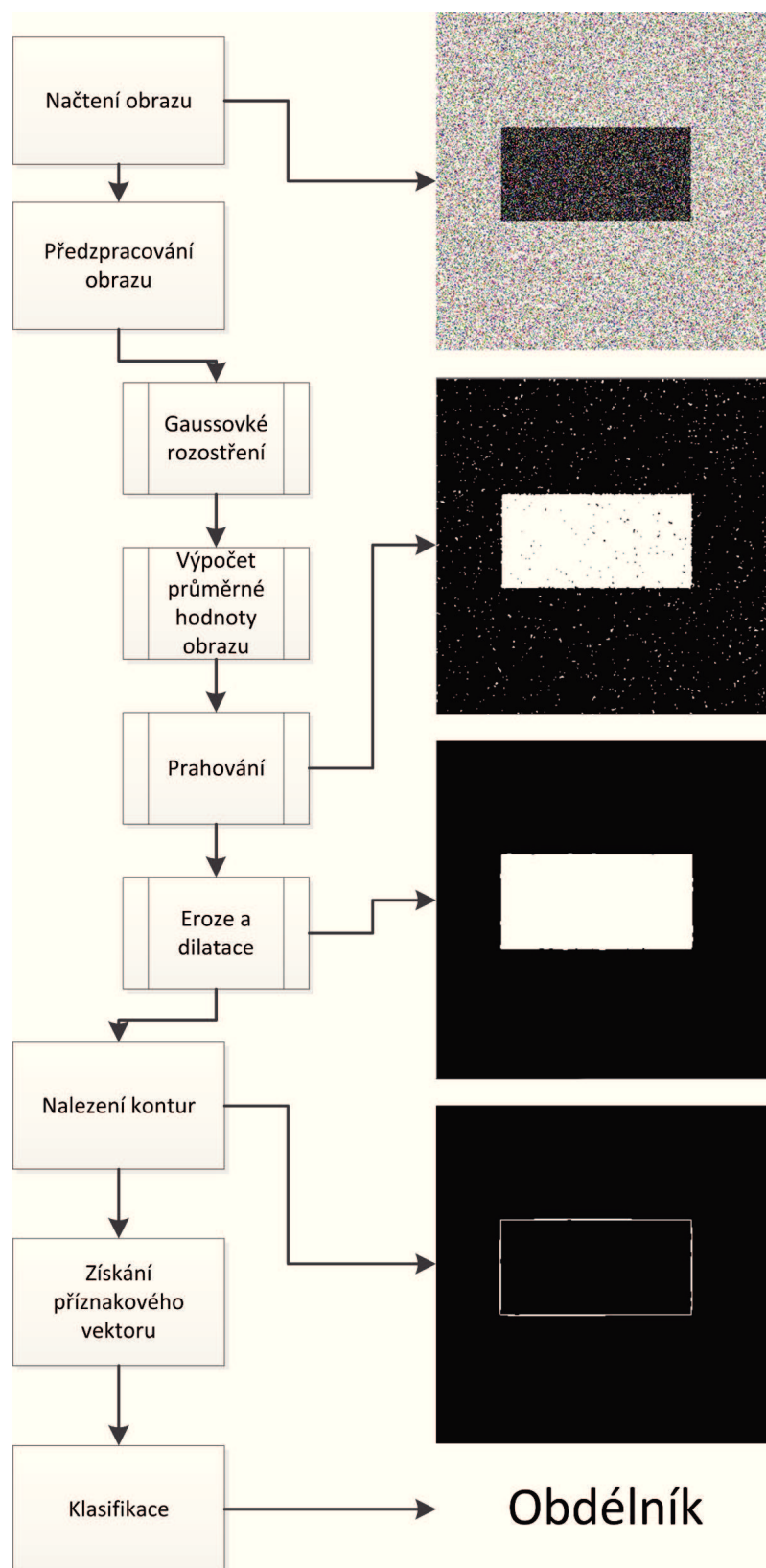
Jelikož aplikace není určena pro nasazení v praxi, ale pouze pro testovací účely v rámci bakalářské práce, je aplikace pouze konzolová a nemá propracované uživatelské rozhraní. Při spuštění se aplikaci jako parametr předá cesta k obrazu, který chceme zpracovat. Program obraz zpracuje a pokusí se určit třídu, do které objekt na obraze patří. Program vypíše předpokládanou třídu objektu a jeho zjištěné vlastnosti.

7.1 Tréninková data

Pro testovací účely jsem v grafickém editoru vytvořil sadu umělých obrázků. Jedná se o jednoduché geometrické tvary (kruh, čtverec, elipsa ...). Jedna část obrázků slouží pro učení rozhodovacího stromu, další část obrázků slouží pro jeho otestování. Dále pak bylo vytvořeno několik obrázků, které slouží pro otestování jednotlivých příznaků. Všechny obrázky, které byly použity, jsou uloženy na přiložené CD v adresáři Obrázky. Funkce `loadImages` nejprve načte tréninkové obrázky podle seznamu ze souborů a poté je předá k dalšímu zpracování.



Obrázek 30: Ukázka několika tréninkových obrázků



Obrázek 31: Diagram aplikace

7.2 Předzpracování

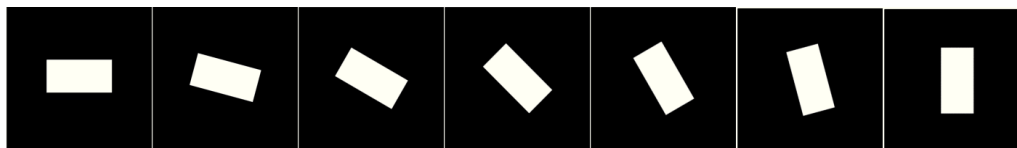
K tomu, abychom mohli získat příznaky jednotlivých objektů, musíme obraz nejprve předzpracovat. Předzpracování provádí funkce `preprocessImage`, jejímž vstupem je barevný obraz. První operací, která se provede, je převedení barevného obrazu na šedotónový. Další operací předzpracování je použití Gaussova filtru pro potlačení vyšších frekvencí, tedy například náhodného šumu. Poté následuje výpočet průměrné hodnoty jasu obrazu a směrodatné odchylky pomocí funkce `cvAvgSdv`. Tyto hodnoty jsou poté použity pro určení prahu T při prahování daného obrazu. Při prahování je pro objekt použita bílá barva a pro pozadí barva černá. Na binárním obraze, který vznikl prahováním jsou provedeny morfologické operace eroze a dilatace. Provedením eroze se částečně zbavíme šumu a zjednodušíme strukturu objektu. Naopak dilatace je využita k zaplnění malých děr v objektu. Pro erozi i dilataci je použit strukturní element velikosti 3×3 pixely.

7.3 Získání příznaků

Poté, co je obraz předzpracován, můžeme přistoupit k získávání jednotlivých příznaků. Získání příznaků provádí funkce `getFeatureVector`, která zjistí jednotlivé příznaky objektu a vrátí je v podobě příznakového vektoru. V rámci popisu objektu je nejprve nalezena kontura objektu pomocí funkce `cvFindContours`, která nám vrací nalezené kontury v dvouúrovňové struktuře, kde v nejvyšší úrovni jsou kontury objektu a v úrovni druhé jsou kontury děr, které se v objektu nalézají. Na základě nalezených kontur jsou pak zjišťovány jednotlivé příznaky a jsou uloženy do vektoru.

7.4 Testování

Po vytvoření rozhodovacího stromu jsem se rozhodl otestovat všechny příznaky a klasifikátor jako celek. K tomu mi posloužila množina 30 testovacích obrázků, které jsem vytvořil. Jelikož se jednalo o umělé obrazy, u nichž nebyl problém s úplnou a velmi přesnou segmentací, a tedy i jednoduchým získáním příznaků, není velkým překvapením, že klasifikátor určil správnou třídu objektu ve 100% případech. Další testovací obrazce prověřují některé konkrétní vlastnosti příznaků. Jedním z testovacích obrazců byl obdélník, který jsem postupně natáčel v rozmezí 0 až 90 stupňů, v každém kroku vždy o 15° . Účelem bylo zjistit, jak jsou jednotlivé příznaky invariantní proti natočení objektu. Výsledky je možné vidět v následující tabulce.



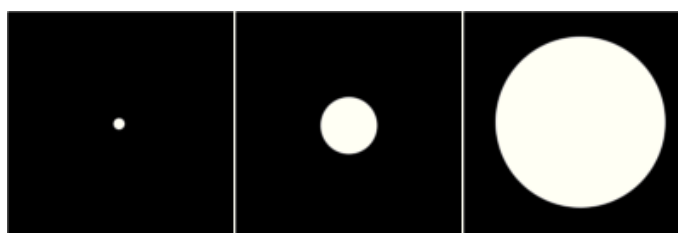
Obrázek 32: Testovací obrazce

Vlastnosti objektu	Obdélník pootočený o						
	0°	15°	30°	45°	60°	75°	90°
Velikost	15 931	16 239	16 212	16 186	16 297	16 239	15 931
Obvod	536	535	535	536	536	535	536
Kompaktnost	18.03	17.65	17.68	17.76	17.63	17.59	18.03
Podlouhlost	2.011	1.993	2.000	2.008	1.993	2.001	2.011
Pravoúhlost	1	0.995	0.998	0.999	0.997	0.996	1
Konvexnost	1	0.995	0.991	1	0.994	0.991	1
Hu1	0.2090	0.2077	0.2083	0.2075	0.2075	0.2082	0.2090
Hu2	0.0159	0.0154	0.0156	0.0153	0.0153	0.0156	0.0159

Tabulka 1: Tabulka příznaků pro různé pootočený obdélník

Jak je vidět na naměřených výsledcích, jsou jednotlivé testované příznaky dosti odolné proti natočení objektu. Jedinou výjimku tvoří velikost, kde naměřený rozdíl byl až 366pxl.

Některé vlastnosti objektů jsou velmi závislé na reprezentaci obrazu. Při nízkém rozlišení jsou zkresleny vlastnosti objektů, které nejsou pravoúhlé nebo jejichž hrany nejsou paralelní s hlavními osami obrazu, což vyplývá z reprezentace digitálního obrazu jako mřížky. Ke zkreslení dochází také, pokud má samotný objekt malou velikost. Toto jsem otestoval na kruzích o různé velikosti (viz Obr.33). Například kruh o velikost 70 000pxl má kompaktnost 12.59, což se blíží teoretické hodnotě 4π (12.566), která by měla platit pro všechny kruhy. Zatímco při velikost 2055 je kompaktnost 12.73 a při velikosti 216 dokonce 13.81, což je hodnota již dosti odlišná. Také příznak pravoúhlosti je zmenšujícím se rozlišením ovlivněn. Kruh s velikostí 216pxl má pravoúhlost hodnotu 0.944. Teoretická hodnota pravoúhlosti je přitom pro kruh $\pi/4$ (0.785). Pro obdélník, jehož hrany jsou rovnoběžné s rastrem obrazu, se hodnoty příznaků mění se zmenšující se velikostí jen minimálně.



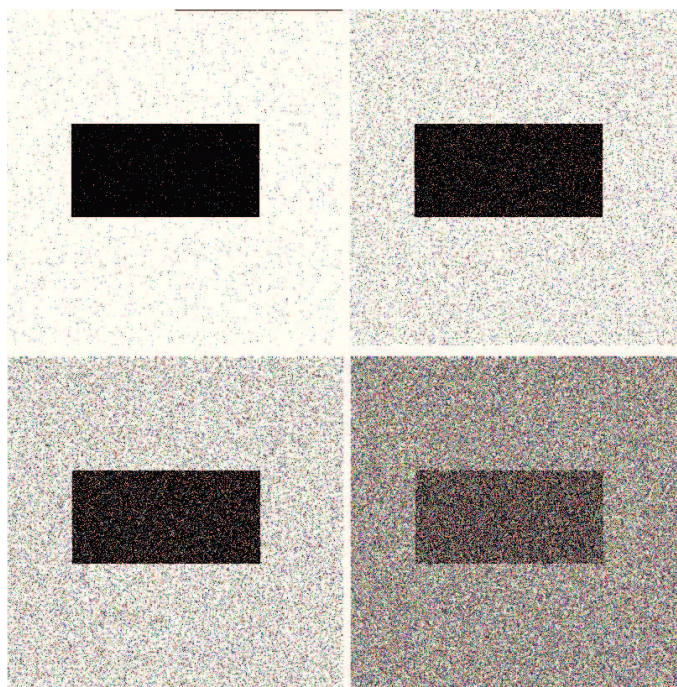
Obrázek 33: Kruhy použité pro testování

Vyzkoušel jsem také, jak bude program reagovat na obrázky, které jsou v různé míře zatíženy náhodným šumem (viz Obr. 34). Program byl schopen ve velké míře šum potlačit, a to především použitím morfologických operací na prahovaném obraze. Při extrémních

Velikost	69 495	7 881	2 055	763	216
Obvod	935	316	161	99	54
Kompaktnost	12.59	12.68	12.73	12.97	13.81
Podlouhlost	1	1	1	1	1
Pravoúhlost	0.789	0.800	0.800	0.848	0.944
Konvexnost	0.997	0.985	0.988	0.974	0.923
Hu1	0.1592	0.1592	0.1592	0.1596	0.1620

Tabulka 2: Tabulka příznaků pro kruhy různých velikostí

hodnotách šumu byly některé vlastnosti objektu zkresleny, nikoliv však do té míry, aby byla ovlivněna klasifikace.



Obrázek 34: Obraz poškozený různým množstvím náhodného šumu

7.5 Zhodnocení

Podle očekávání příznaky velmi dobře posloužily pro rozpoznávání geometrických tvarů v sadě uměle vytvořených obrazů. Tyto obrazy bylo totiž možné velice dobře segmentovat a bylo tedy možné i velice přesně určit kontury objektů, z kterých jsou poté vypočteny hodnoty příznaků. Výsledky testů byly částečně zhodnoceny již výše, zde uvádím určité shrnutí zjištěných výsledků. Veškeré testované příznaky, s výjimkou Eulerova čísla, jsou založeny měření geometrických vlastností a je poměrně logické, že jsou závislé na přesné

segmentaci a také, že nejsou odolné proti geometrickým transformacím obrazu. Toto je také asi největší slabinou vybraných metod. Mezi výhody naopak patří invariantnost proti afinním transformacím, poměrně snadná implementace a velmi dobrá interpretovatelnost pro člověka.

8 Závěr

V teoretické části práce byl podle zadání popsán řetězec zpracování obrazu, počínaje získáním obrazu přes předzpracování, extrakci příznaků až po klasifikaci. Zvýšený důraz byl přitom kladen právě na příznaky a příznakové rozpoznávání. Vzhledem ke složitosti a rozsahu problematiky týkající se příznakového rozpoznávání však nebylo možné v práci obsáhnout všechny metody, postupy a algoritmy. Proto byly vybrány a popsány pouze některé z nich.

V praktické části byla vytvořena aplikace, která slouží k testování vybraných příznaků. Rozhodl jsem se použít několik radiometrických příznaků a sadu sedmi geometrických invariant tzv. Hu momentů. K předzpracování a extrakci příznaků program využívá funkce naimplementované v knihovně OpenCV a ke klasifikaci je použit binární rozhodovací strom. Kromě programu samotného byla také vytvořena sada tréninkových a testovacích obrázků. V příloze A je ukázáno několik z těchto obrázků společně s příznaky, které u nich byly zjištěny. Aplikace je společně s výsledky testování popsána podrobně v kapitole 7.

Při testování jsem ověřil funkčnost a použitelnost a také vlastnosti vybraných příznaků. Jednalo se především o odolnost proti otočení, posunutí, změně měřítka, dále také odolnost proti náhodnému šumu a špatné segmentaci. Testované příznaky by bylo možné použít jako základ reálné aplikace, sloužící například pro rozpoznávání psaného textu, které může být použito pro snímání adres z poštovních zásilek a jejich následnému třídění na základě zjištěných informací nebo čtení registračních značek automobilů, dále by bylo možné uplatnění například při převodu psaných textů do digitální podoby apod. Obecně se radiometrické příznaky hodí spíše pro aplikace, kde můžeme zajistit konstantní polohu kamery vůči snímanému objektu, a tudíž nedochází ke geometrickému zkreslení, které by mohlo ovlivnit klasifikaci. Abychom dosáhli uspokojivých výsledků bylo by pravděpodobně nutné zvolit sofistikovanější metodu segmentace, než byla použita ve vytvořeném programu. Kromě výše uvedeného by bylo jistě možné najít nebo vymyslet spoustu dalších úloh, kde by se příznakové rozpoznávání dalo uplatnit. Zde jsem však uvedl pouze několik ilustrativních příkladů.

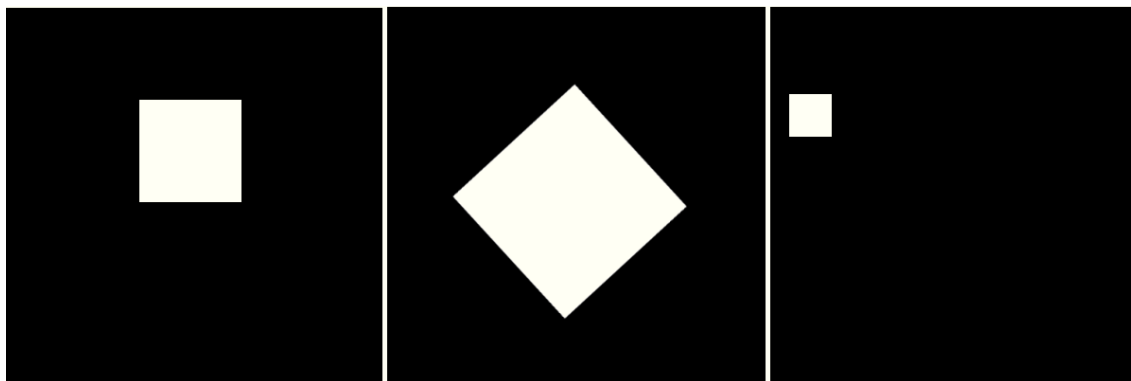
9 Literatura

- [1] BRADSKI, G., KAEHLER, A. *Learning OpenCV* Sebastopol: O'Reilly, c2008, 555 s. ISBN 978-0-596-51613-0.
- [2] HLAVÁČ, V., ŠONKA, M. *Počítačové vidění*. 1. vyd. Praha: Grada, 1992, 272 s. ISBN 80-854-2467-3.
- [3] SOJKA, EDUARD *Digitální zpracování a analýza obrazů*. 1. vyd. Ostrava: VŠB - Technická univerzita, 2000, 133 s. ISBN 80-707-8746-5.
- [4] GABRIEL, P. *Klasifikace objektů v obrazech*. Brno, 2009. 78s. Diplomová práce VUT. Vedoucí diplomové práce Ilona Kalová.
- [5] *Wikipedia* [online] Dostupné na adrese <<http://www.wikipedia.org/>>
- [6] PETYOVSKÝ, P. *Reprezentace a vlastnosti obrazových dat* [online] Dostupné na <<http://www.uamt.feec.vutbr.cz/vision/TEACHING/MPOV/mpov.html.cs>>
- [7] KALOVÁ, I. *Segmentace a detekce geometrických primitiv* [online] Dostupné na <<http://www.uamt.feec.vutbr.cz/vision/TEACHING/MPOV/mpov.html.cs>>
- [8] KALOVÁ, I. *Regionální segmentace a shlukování* [online] Dostupné na <<http://www.uamt.feec.vutbr.cz/vision/TEACHING/MPOV/mpov.html.cs>>
- [9] HORÁK, K. *Popis objektů* [online] Dostupné na <<http://www.uamt.feec.vutbr.cz/vision/TEACHING/MPOV/mpov.html.cs>>
- [10] HORÁK, K. *Matematická morfologie* [online] Dostupné na <<http://www.uamt.feec.vutbr.cz/vision/TEACHING/MPOV/mpov.html.cs>>
- [11] *Wolfram MathWorld* [online] Dostupné na <<http://mathworld.wolfram.com/>>
- [12] *Předzpracování obrazu* [online] Dostupné na <<http://www.uamt.feec.vutbr.cz/vision/TEACHING/MPOV/mpov.html.cs>>
- [13] *Technical-itch* [online] Dostupné na <<http://technical-itch.co.uk/2008/08/20/digimemo-a402-digital-notepad-review/>>
- [14] *Rob Faludi* [online] Dostupné na <<http://www.faludi.com/2008/03/20/summer-xbee-course-sociable-objects/>>
- [15] *Technet* [online] Dostupné na <http://i.idnes.cz/12/023/c15/VSE417861_GoogleAuto.jpg>
- [16] *Scarlet Workflow* Dostupné na <<http://www.scarletuser.com/showthread.php?t=567>>

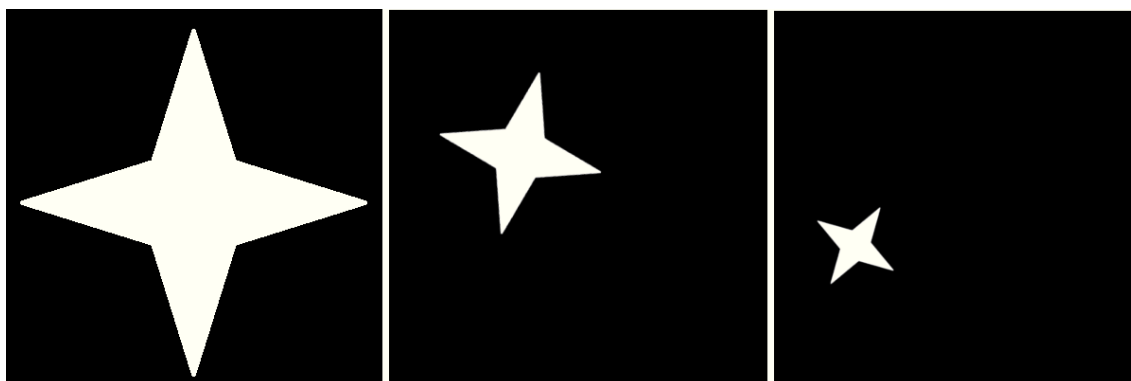
- [17] *Signal Image Processing Lab*[online] Dostupné na <<http://sipl.technion.ac.il/>>
- [18] *Rozhodovací stromy* [online] Dostupné na <<http://cmp.felk.cvut.cz/>>
- [19] *Chain Codes* [online] Dostupné na <http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MARBLE/medium/contours/chain.htm>
- [20] KALOVÁ, I. *Předzpracování a segmentace* [počítačový soubor] FEKT

10 Přílohy

A) Ukázky obrázků a k nim zjištěných příznaků



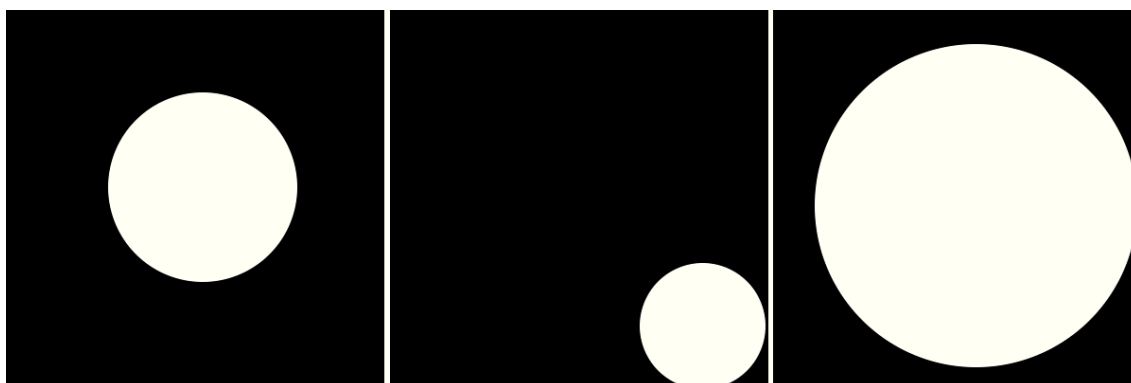
Velikost	11 449	30 299	2026
Obvod	428	691	180
Kompaktnost	16.00	15.78	16
Podlouhlost	1	1	1
Pravoúhlost	1	0.999	1
Konvexnost	1	0.992	0.958
Hu1	0.1667	0.1667	0.1667



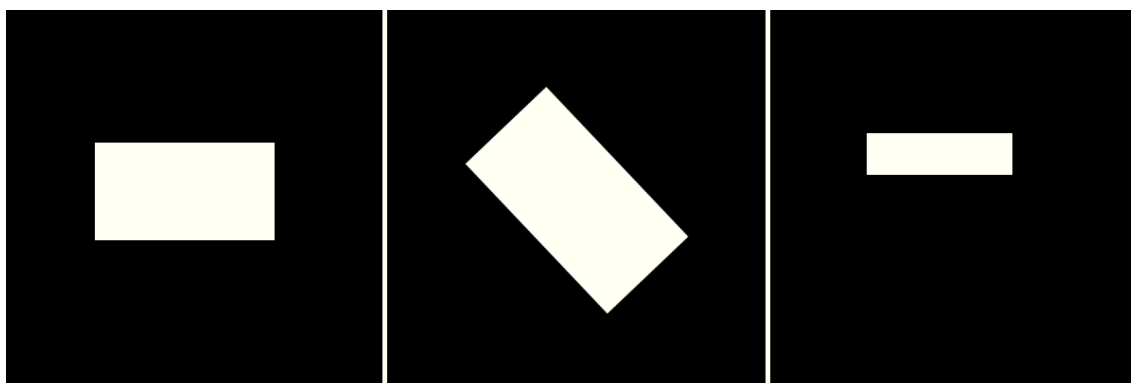
Velikost	33 303	7 877	2 587
Obvod	1168	554	306
Kompaktnost	40.96	39	36.27
Podlouhlost	1	1	1.01
Pravoúhlost	0.492	0.505	0.544
Konvexnost	0.492	0.508	0.545
Hu1	0.2364	0.2343	0.2328



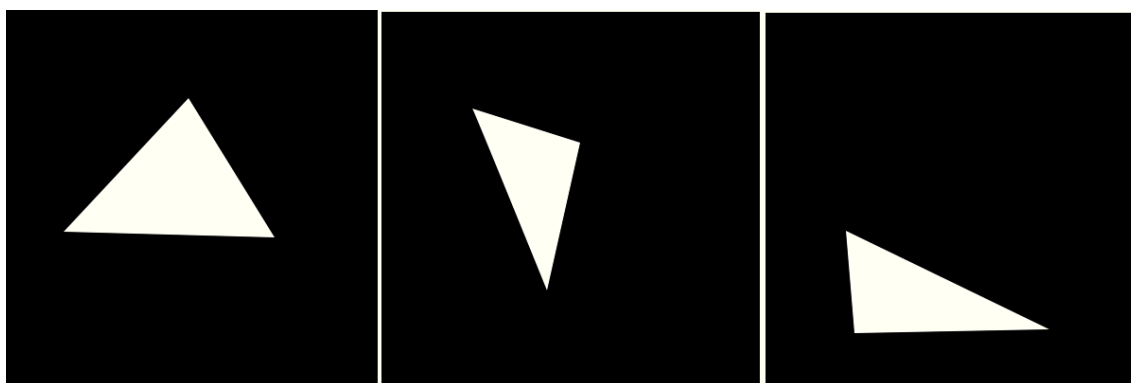
Velikost	14 330	11 080	22 516
Obvod	493	525	538
Kompaktnost	16.97	24.88	12.85
Podlouhlost	2.506	4.33	1.25
Pravoúhlost	0.791	0.788	0.789
Konvexnost	0.989	0.987	0.996
Hu1	0.231	0.360	0.163



Velikost	31 103	13 835	89 092
Obvod	626	418	1059
Kompaktnost	12.61	12.65	12.60
Podlouhlost	1	1	1
Pravoúhlost	0.794	0.805	0.792
Konvexnost	0.995	0.994	0.995
Hu1	0.1591	0.1591	0.1592

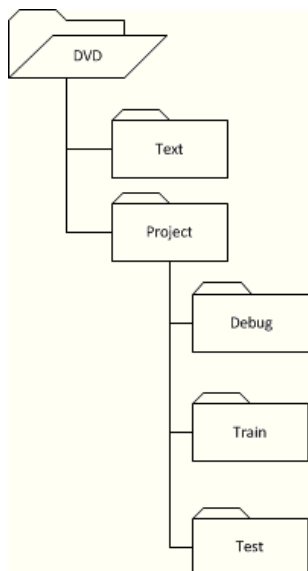


Velikost	19 278	25 490	6 967
Obvod	582	665	395
Kompaktnost	17.57	17.35	22.43
Podlouhlost	1.853	1.851	3.444
Pravoúhlost	1	0.999	0.999
Konvexnost	1	0.990	1
Hu1	0.1993	0.1991	0.3109



Velikost	16 077	9 560	11 160
Obvod	580	480	543
Kompaktnost	20.99	24.11	26.45
Podlouhlost	1.531	2.217	2.514
Pravoúhlost	0.514	0.512	0.513
Konvexnost	0.989	0.983	0.985
Hu1	0.2015	0.2430	0.2779

B) Obsah DVD



Text Obsahuje originál bakalářské práce, zadání práce a čestné prohlášení.

Project Obsahuje vytvořený projekt v prostředí Microsoft Visual Studio 2010 se zdrojovými kódy.

Debug Obsahuje spustitelnou aplikaci. Spouští se souborem ImageOpenCV.exe

Test Obsahuje sadu vytvořených obrázků určených k trénování klasifikátoru.

Train Obsahuje sadu vytvořených obrázků určených k testování.